DOCUMENT RESUME

ED 108 598 IR 002 109

AUTHOR	Follettie, Joseph F.
TITLE	Experimental and Student-System Interactive Instructional Illustrations Pertinent to IDCMS.
INSTITUTION	Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO	SWRL-TN-1-72-01
PUB DATE	4 Feb 72
NOTE	23p.
EDRS PRICE	MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS	Audiovisual Aids; Computer Programs; *Computer Science; Computer Storage Devices; Educational Development; *Educational Experiments; *Information Systems; *Instructional Technology; Interaction; Learning Processes; Program Design; Skill Analysis; *Systems Analysis; Systems Approach
IDENTIFIERS	IDCMS

ABSTRACT

Southwest Regional Laboratory IDCMS (Instructional Development Control and Monitoring System) will be a flexible hardware system for controlling and monitoring instruction and research in the laboratory setting. This paper seeks to introduce potential users to the system and software designers to representative challenges that system exploitation will pose. (Author/SK)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED108598

SWRL

SOUTHWEST REGIONAL LABORATORY TECHNICAL NOTE

DATE February 4, 1972

NO TN 1-72-01

SCOPE OF INTEREST NOTICE
The ERIC Facility has assigned this document for processing to: IR SP
In our judgement, this document is also of interest to the clearinghouses noted to the right. Indexing should reflect their special points of view.

TITLE: EXPERIMENTAL AND STUDENT-SYSTEM INTERACTIVE INSTRUCTIONAL ILLUSTRATIONS PERTINENT TO IDCMS

AUTHOR: Joseph F. Follettie

ABSTRACT

SWRL IDCMS will be a flexible hardware system for controlling and monitoring instruction and research in the Laboratory setting. This paper seeks to introduce potential users to the system and software designers to representative challenges that system exploitation will pose.

IR 002 109

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 11300 La Cienega Boulevard, Inglewood, California, 90304.

EXPERIMENTAL AND STUDENT-SYSTEM INTERACTIVE INSTRUCTIONAL ILLUSTRATIONS PERTINENT TO IDCMS

Joseph F. Follettie

The SWRL Instructional Development Control and Monitoring System (IDCMS) is designed to provide broad support for the Laboratory mission. That is, it will be responsive to a variety of specific organizational requirements. This paper will deal only with the system's capability for supporting requirements for a) conducting educational research and b) for trying out instruction that stems from such research. Comments on instruction will be further restricted to those sorts of instruction that feature nontrivial levels of student-system interaction. While certain details of the illustrations to be offered may appreciably challenge system software designers or even lie beyond capabilities of the system in Version 1 configuration, the intent is only to promote a better understanding concerning how the interactive domain will be bounded during earliest applications of the system. The illustrations are offered as conjecturally consonant with system capabilities as these can be read from preliminary documentation. If facets of the illustrations go beyond what the system in Version 1 configuration will be able to do, then it will be useful to find out in what respects this is so and whether such features can be obtained quickly and cheaply consonant with earliest usage of the system--that is, usage during the first semester of the 1972-73 school year.

Although it is inevitable that the system will not be able to do certain things at the outset that users would like it to do--or, more probably, not do them with sufficient ease--preliminary documentation suggests that system capability for supporting interactive mode educational research and tryouts of instruction will fall somewhere between one and two orders of magnitude beyond the pre-system Laboratory capability for supporting such efforts. General understanding of system support capability among instructional design-development staffs should be followed almost immediately by a dramatic rise in productivity of the efforts of these staffs. However, the system will be little exploited until such understanding prevails.

This paper seeks to stimulate both prospective users and software designers; for it is necessary that the two groups reach a common understanding concerning support capabilities that it will prove reasonable to assume that the system in Version 1 configuration can supply. Student-system interactive instruction was selected as a first query domain because it is one that will tax any system, will encompass much work that has needed doing for years, and likely will produce the greatest number of misperceptions concerning system functions and system capabilities.

Hardware Configuration

The system is controlled by a NOVA 1200 device having, in Version 1 configuration, a primary storage of 16 K (16,384) 16-bit words that can be assigned in different proportions to Read Only (program) and Read-Write (e.g., analysis functions) memories. Should it prove necessary, the controller can be core-to-core interfaced with larger computer configurations. (Although such interfacing currently is viewed as off-line, it is assumed that on-line coupling could be effected if nontrivial levels of on-line calculations underlying conditional advance are required. One such on-line analysis might apply sequential analysis technique to progressive analysis of a sequence of responses to test items, with the test terminating after n items are administered or whenever upper or lower decision boundaries are pierced, whichever occurs first. Should it be required that such analyses occur under a time-share condition wherein each of 6 Ss responds once every 6 seconds--1 response and 1 recomputation per second on the average--then it appears that further computer power would have to come into the system on an on-line basis.)

The following system characterization comments address users who may have little or no awareness concerning hardware assemblies of the system. A good place to begin is with the system's analog-form video disc storage--approximately 1800 frames (32 are lost to system control). This store can be random accessed at acceptably-high speed (for most purposes). The accessed frame goes to S's own video buffer for still display of any duration on S's CRT. Should a second S require the same frame at almost the same instant, he would have to wait a short while (typically no more than a second that is additional to frame accessing-copying time, itself typically less than a second). Video frames are off-line loaded using Television System equipment (not a part of the IDCMS contract).

The audio system consists of an audio loader, extensive audio storage on tape (in 12 tape transports, in Version 1 configuration) and audio buffers for copying specified audio programs from the audio store. An interesting feature of audio tape storage of programs is that intrinsic digital codes controlling selection-synchronization of video frame displays can be formed on the audio tape using a pulse-modulated 55 Hz subcarrier. A digital encoder permits off-line coding of tapes. Master tapes use one track for the digital codes and another for audio messages, thus permitting recoding and other editing of masters. Code and message tracks are merged for programs in audio storage and buffers.

Two data recording systems are available--a digital disc system for keyboard responses and a digital tape system for voice responses. Both have their own control units, which in turn are controlled by the NOVA 1200 controller. The digital disc system has a capacity in excess of 1.2 million 16-bit words. It is evident that not all of this capacity need be used for recording data. Capacity of the digital tape system for IDCMS in Version 1 configuration is modest but expandable. The digital

tape system will not become maximally useful until a speech synthesizer becomes part of the system. This currently is not included due to cost balanced against state of the art.

User-referenced software is loaded into the system through a teletypewriter which encodes instructions via a paper tape punch device. Essentially nonorganic augmentation of audio-video presentation capability can be effected using external-source audio and video material which enters the system through patchboards. System outputs referenced to user responses, latencies, and the like have not yet been identified (by me, that is).

The Version 1 system contemplates supporting instruction or research which employs up to 6 Ss simultaneously.[1] Each S is provided with a terminal whose principal features are a 12" Trinatron color CRT, a set of headphones, and a response keyboard. Other input-output devices can be readily "jacked" at a terminal. Depending on applicable characteristics of controlling software, the system can be caused to vest certain interactive options in S. (One that it appears to vest routinely in S--or in E standing beside S--is the item advance option. Desirably we will be able to vest such an option either in S or in the system and, if in the system, either conditional on S's prior performance or not.)

Finally, the system incorporates a character generator (also a part of the Television System and procured through that contractor) that can be used to caption video displays. The system apparently also easily can be made to accommodate a line generator or whatever is needed to cause x, y performance plots--e.g., plots that contrast some facet of S's performance over time with criterion performance--to come up on command.

Interactive Instruction

It is possible to characterize interactive instruction and interactive educational research within the same framework. That is, different groups can receive different versions of a given instructional program, with variations across versions tapping one or more factors of research interest. The illustration to be presented is at a level of complexity that is consonant with ready extrapolation to the research domain. The point of view of this paper is that by the time we are ready

---

[1]Illustrations to be presented will assume that up to 6 Ss time-share the system while participating in the same instructional program or experiment. The alternative--running up to 6 instructional programs or experiments simultaneously--entails assigning portions of core to each program or experiment. It is apparent that this can occur only if time-sharing instructional programs or experiments (or a mix of the two) can get along with proportionately reduced core.

to engage the system in small-scale tryouts of interactive instruction, we will have laid the groundwork for that instruction in consequence of extensive interactive educational research. The instructional disguise of the illustration, therefore, is intended to be penerated.

Gross Anatomy of Interactive Instruction. Although the contractor has characterized student-system interaction as the simple-minded sort that went out of vogue over a decade ago--i.e., as referenced to a one-response decision domain--IDCMS is potentially much more powerful than the contractor's view implies. To begin with, mainline versions of any instructional program can be viewed as linear progressions. We will make do here with just one mainline version of a program. It will have a start point, followed by a linear progression of lessons and decision points, with exit--e.g., to following program--through a terminal decision point. The decision point permits E (or the system) to evaluate a given skill using 2-20 items, to diagnose the cause of unacceptable performance (whether referenced to a current decision point or to a sequence of decision points culminating in the one now visited by S), and to prescribe instruction designed to raise unacceptable performance to an acceptable level.[2] If S responds as intended at each decision point, then he will negotiate the program along the instructional path traced by the mainline version of the program.

There are as yet no acceptable conventions for labelling the alternatives to negotiating the program along the instructional path traced by its mainline version. We are concerned here with a requirement that every S in time exit through a single terminal decision point to whatever may follow. Worth noting in passing, a program that truly branches precludes advancing all Ss through the same terminal decision point. True branching programs offer S a choice, at one or more decision points, concerning whether he will travel to a specified terminal decision point or to an alternate terminal decision point. A program that truly branches does not ever converge (at least in any finite practical sense) its branches, once these are formed. Hence, references to branching in an instructional context cannot signify true branching when the intent is to advance all Ss through the same terminal decision point. The apparent forms of departure from a mainline linear progression when the objective is a common exit are loops, skips, extended alternate paths, and combinations of these.

Two sorts of loops may be distinguished. The first--a simple recycling loop--requires S to return to an earlier mainline lesson in consequence of diagnostic-prescriptive outcomes experienced at a decision

---

[2]See Ronald Besel's TN-5-72-04, "Diagnosis-Prescription in the Context of Instructional Management," for definitions of diagnosis and prescription.

point occurring somewhere on the mainline instructional path. The recycling loop in pure form seems pedagogically unattractive. It requires S to renegotiate the lesson in exactly the same form it was negotiated before. Apart from the unacceptable rote-memorization implications of such a procedure, we are stuck with the fact that earlier instruction proved ineffective for the particular S under consideration. Diagnosis should indicate what modifications are in order; prescription should indicate the address of suitably-modified instruction addressing the same program skill as the failed mainline lesson. We return to such alternative instruction in a moment.

The second sort of loop might be called an extra-program loop. All instruction presumes certain prerequisite skills occur in S's skills repertoire. These typically are called entry skills. S moves forward from a program start point only if the assumption appears tenable that he possesses the entry skills on which the program is predicated. While such assumptions can be evaluated to a degree, it is impossible to foresee and evaluate every sort of prerequisite skill that might apply, although informal evaluation of major approximations is possible. We assume here that extra-program loops reference to the major perceived entry skills and that initial tenability of an assumption that S possesses such skills ought not close out on the possibility that later, more-analytic evaluation will show holes in his entry skills repertoire. Where a hole is detected at a given decision point, it is possible that all that will be required is that S loop to supplemental instruction addressing the deficient entry skill, with exit from the loop being back to the decision point at which the loop originates.

If S loops back to an earlier program skill but not to the same lesson he earlier negotiated--but rather an alternate lesson, perhaps one that elaborates on what was presented earlier--then the loop shares certain characteristics of the recycling loop and of the extra-program loop. That is, he recycles to a program skill for which he received earlier instruction but he does not simply recycle to the same lesson but rather to an alternative lesson. We characterize this departure from a linear progression as a mixed loop departure.

Mainline linear progressions that--installed in the schools--permit skips probably require redesign. However, experimental mainline linear progressions might either accept skips as an undesirable interim phenomenon that can be lived with or as a phenomenon whose effects upon acquisition are worthy of study in their own right. A skip occurs when performance at a given decision point indicates that S should skip one or more later lessons in the progression. Setting aside rest-review-refresher lessons, a skip decision could only be reached if testing at a given decision point references both to skills negotiated earlier and skills yet to be treated. It appears tenable that a skip most often will occur when only logical semiordering of earlier lessons applies and S experiences a mixed loop departure that carries him back to the rear-most of two or more lesson sequences addressing skills at the same

level in a proficiency hierarchy. If these program skills are independent in the sense that either is treatable independently of treating the other, then a mixed loop departure might be followed by a skip that returns S to the decision point wherein unacceptable performance originally was detected. This would constitute a mixed loop plus skip form of departure from the mainline progression.

When a mixed loop retrogression is followed by progression along an alternate path to the mainline path until the original decision point departed from is reached or when S proceeds from such a decision point forward by an alternate path that closes with the mainline path at a terminal or intervening decision point, then the departure might be characterized as an extended alternate path departure. Particularly when such departures are in the forward direction, some are tempted to call them branches, even though it is clear in light of the earlier discussion that they do not represent true branches. Alternate paths might arise for a variety of reasons--e.g., differential demands across Ss for elaboration on instruction, as revealed by accuracy and rate of response on achievement or diagnostic tests. (A decision to assign S to a given alternate path is not irrevocable. The decision might be reevaluated at each decision point along the progression, with reclassification a continuing option.)

The variations in interactive instruction referenced to a mainline version of such instruction take the form primarily of departures from the mainline linear progression of the following types: extra-program loops, mixed loops, mixed loops with skips, simple skips forward, and extended alternate paths, whether or not following retrogression. The anatomy of interactive instruction--at the skeletal level if not the histological-cytological--is that which is afforded by these options for forming departures from mainline instruction conditional on performance. In the current era we do not imagine a system of the IDCMS type using artificial intelligence capabilities for perceiving useful departures from mainline instruction or for acting on such perception to form suitable departure materials. Another way to put this is that E, during the current era, will identify all entertainable forms of departure, prepare materials sets that can be used under given departure conditions, prepare all achievement, diagnostic, and other material sets that are of interest, and do whatever else is required before IDCMS can be asked to do its thing. This does not preclude the probe model made famous by operant conditioners, wherein several treatments in succession are tried out on the same S, so long as it is understood that E must foresee all of the alternative treatment sets he will study under such conditions and so appropriately arm IDCMS.

As noted earlier, specification of the anatomy of an interaction system that might be used to tryout instructional programs can be considered as first step in specification of interaction research. Thus we can use an instructional program having all of the interactive characteristics--dependent on characteristics of S's performance--enumerated as a base view of an interactive instructional system. We might then study the performance of different groups of Ss when

given base form instruction is administered but pacing options, response characteristics used to define path-conditional instruction, and the like are varied across groups. We reaffirm, then, that illustrative interactive instruction simply needs to be carried a step or so forward to become illustrative interactive instructional research.

An Interaction Scenario for Spelling Instruction. Short instructional programs preclude formulating interactive research that is much interactive. Conversely, extended instructional programs tend to imply interactive research whose hardware requirements cannot yet be met or can be met only at prohibitive cost. Instructional programs of intermediate length permit formulation of interactive research that is subroutine rich (and so promising) while making only reasonable demands on hardware in light of state of the art and available resources. Moreover, system-referenced multiple user employment of an interactive system addressing instruction of intermediate length has the realistic effect of raising time-pressured storage-retrieval burdens--a time-share facet of the system--above trivial levels, since a system that is responsive to student rate predispositions must in time cause students to spread out over an appreciable stretch of the program. For these reasons, we assume a unit of instruction that a child of instruction-referenced median rate will complete in 15 30-minute periods--e.g., in 3 weeks, 5 days a week, 30 minutes a day.

So that we can get instruction far enough along to warrant extensive generalization responses based on prior acquisition of rules of an instructional knowledge system, we assume that the illustrative program unit is the third such in an ordered program of spelling units. We assume further that children entering this third unit of spelling instruction have similar entry skills profiles, although perhaps varying in chronological age and educational experience, including time spent on the first two spelling units.

One could considerably unburden the system regarding on-line interaction simply by having children sit around whenever the system found it convenient for them to do so. The pony express and bicycle messenger "interactive" systems manifest this predisposition toward "off-line interaction." It is possible that student-generated demands for breaks in instruction will reduce usership, thus lowering the on-line burden placed on the system. However, system-generated demands for "breaks" can be tolerated only as a last-resort concession to state of the art, since the object is to increase educational productivity through automation. This can hardly be done by sacrificing student instructional time just to please the machine. The objective is to optimally speed instruction consonant with student characteristics. Agreeing that we cannot outstrip state of the art, we must ask the machine to be a part of the solution, rather than allowing it to be part of the problem. It tends to be true for all such systems, and promises

to be particularly true for IDCMS, that the hardware will not stand in the way of a broad field of sophisticated uses of the system. Foregoing remarks of this paragraph could be interpreted to say that we contemplate fully challenging the ingenuity of software people.

Gross characterization of the anatomy of interactive instruction reveals that the system will react to a response sequence--or even to a set of such sequences--rather than to a single response to a single test item. Its reaction to a decision formed on a response sequence will be at the same level. That is, following a decision, the system will direct S to a lesson--or to an instructional sequence--rather than to a single instructional or test item. This might be contrasted with the item-to-item interactive models introduced over a decade ago by programmed instruction researchers. The item-to-item model assumes two things that do not seem reasonable: a) that a single response to a single item is a reasonable basis for evaluation, diagnosis, or prescription, and b) that the fact that S is wrong at the same time signifies why he is wrong. It may be a little slower going, but one can advance through Crowder-type programs simply by throwing darts at response alternatives; it is like playing a slot machine that comes up only 0 or 1, with 0 signifying loop and 1 signifying advance. The spelling program to be sketched will be interactive on a segment-to-segment, rather than an item-to-item, basis. We will characterize segments as of 2-20 "item" length, with abrupt termination of the segment a possibility whenever its items require item-to-item responses through the keyboard and on-line evaluation of performance.

IDCMS permits student-system interactive studies using one or a combination of the following presentation modes: Video Only (VO), Audio Only (AO), Audio + Video (AV). Spelling instruction entails that many or most audio elements have durations on the order of 1 second--the time it takes to pronounce words of one- or two-syllable length. It will be assumed that the illustrative spelling unit for the most part deals with spellings of one-syllable words. The illustrative program will make use of AV items during instructional segments and AO items during test segments (but with complications to be revealed). Some or most instructional items will require keyboard responses. Responses to instructional items will be evaluated from a standpoint of appropriateness of pacing and to determine whether perceptual-mechanical skills assumed brought to instruction are really there. Responses to test items will be evaluated against spelling criteria (which may differ depending on whether the response refers to a rote-memorized program word or to a rule-generalized novel word and on such factors as point in the instruction to which the criterion will be applied).

Descriptions of IDCMS suggest that pace of interactive instruction will be vested in E or S at the terminal. Software design should make it possible alternatively to vest instructional pace in the system (under software control). The notion that the system should be responsive to student rate manifestations is sound. However, the notion

that this matter should be left entirely in S's hands appears less compelling. The system can be used to challenge S in the pacing domain while at the same time being responsive to his rate manifestations as revealed by a germane performance record. The illustration assumes that the instructional pace is vested in the system, conditional on S's performance. Vesting pace control in the system makes it possible both to keep pace appropriateness in view throughout instruction and to challenge S to reach higher rates as it becomes evident that he could.

A segment of instruction typically will entail presentation of 2-20 AV items. While ordering of such a series may or may not be required, any first presentation of the series could be presented in a specified fixed order without doing harm to pedagogical intent. This suggests several alternatives to composing sequences from single audio and single video elements. Setting aside the use of external sources for presenting fixed-sequence segments because such use implies developing random accessing assemblies that service these sources (which enter the system through patchboards), the major remaining implication is that the system's audio subsystem may have to deal only with longer program sequences, even though most audio elements used will be quite short. Later remarks explore implications of storing fixed-sequence segments in audio store for purposes of supporting the illustrative interactive instruction.

The typical view of instruction is that instructional segments, or lessons, alternate with test segments, or decision points, with evaluation, diagnosis, and prescription confined to the test segments. All know, however, that any instructional segment or item therein may invite all sorts of responses--both covert and overt. If one wishes, one may evaluate what S's eyes do during presentation of instructional items. One may evaluate S's demand rate for item presentation. Or one may define and evaluate a nontest segment response that reflects elements of skill underlying skilled test responses. Illustrative instructional segments will present AV pairs one at a time and require imitation (or copying) responses referencing to the V element during each presentation. During or following presentation of such a segment, the system will evaluate the series of responses against a criterion for acceptability. If unacceptable, the system will select a most-entertainable hypothesis concerning cause, based on special instructional software. This hypothesis will be tested and, if found tenable, will result in presentation of a suitable prescriptive segment of instruction--2-20 items, possibly of AV form. Prescriptive segments referenced to instructional segments typically loop to skills assumed present on entry but found in time not to be.

A test segment also typically might entail presentation of 2-20 items, but of the AO type. Such a series need not be completed if a statistical decision can be reached beforehand indicating availability of needed decision information. First presentation of a given test segment featuring program words typically might be in any predetermined fixed order; later presentations of the segment should be in some other order. AO items would be presented one at a time. S would be required

to construct spelling responses referencing to the A element after each presentation. During or following presentation of such a segment, the system would evaluate the series of responses against a criterion for acceptability (perhaps one criterion for program words and another for novel rule words). If unacceptable, the system would select a most-entertainable hypothesis concerning the cause, based on special instructional software. This hypothesis would be tested and, if found tenable, would result in presentation of a suitable prescriptive segment of instruction. Prescriptive segments referenced to test segments typically will reference to earlier instruction, perhaps extending it in depth or specificity. These segments loop back to instruction addressing program skills but elaborate on earlier instruction as test-referenced diagnosis may require.

The illustrative instructional-evaluative-diagnostic-prescriptive scheme references to rate of presentation and to multiple skills. Below it is outlined in somewhat greater detail.

1. Present Instructional Segment 1. Collect imitation performance data. Evaluate data in light of a given field of hypotheses, where conditions for entertaining each are prespecified. H_0 asserts that performance is acceptable. If H_0 is tenable, S advances to Test Segment 1. Assume H_0 is rejected and that the data suggest the cause is H_1--instructional pace is too fast. If data reveal all or most responses are correct as far as they go but incomplete, then H_1 is tenable and Instructional Segment 1 is presented again at a slower pace. Assume H_1 also is rejected, but that data suggest tenability of H_2--performance is unacceptable due to an information-processing or mechanical (key-pressing) deficiency. All sorts of such deficiencies can be imagined. Let us suppose that the child typically is failing to supply the next to last letter in the word whose letter sequence is to be copied--e.g., CT for CAT, TRP for TRIP, SHP for SHOP.

2. Present Diagnostic Segment 1.1.0 (perhaps rate-split to reflect former and reduced rates). If imitation performance data replicates the Instructional Segment 1 data (even though establishing that omissions extend also to consonants in next to last letter position), then H_2 is accepted. Let us so assume.

3. Present Prescriptive Segment 1.2.0. This instruction causes the child to attend to all letters in the sequence. If the word is TRIP, this might be accomplished by blocking all keys except T at the outset. Only when T is pressed does R become unblocked (and T blocked). So it goes through the sequence of keypresses. Advance to the next item occurs following the pressing of P. This segment might be terminated either when an acceptability criterion for the overall response is met or when interpress response latencies approach uniformity.

4. Present Instructional Segment 1 again. Collect imitation performance data. Assume that data reveals acceptable performance.

5. Present Test Segment 1. Collect spelling performance data. Two sorts of test items might be distinguished: a) those that test for spelling of program words (words whose spellings are copied during instruction), which could be accomplished strictly on the basis of retrieving spellings from memory, and b) those which test for generalization of spelling rules by using as test items novel words (words not used during instruction) that are consonant with spelling rules thus far introduced. Unacceptable performance referencing to program and to novel rule words should have different prescriptive implications.

Fastest and Average Item Consumption Rates. What value should the system evaluation-command-switching sequence of intervals have? The question is relative. Sometimes 5-20 seconds could be allowed; sometimes such delay would be unacceptable. The value of delay between presentations can be defined on the ratio of up to down time. Up time here signifies the time spent by the system presenting items and accepting responses; that is, up time is presentation-response time. Down time is student-handsitting time, used by the system to evaluate the situation, issue commands, and execute these commands (typically by accessing items). The illustrative instructional domain (e.g., spelling, phonics) invites few presentations of speech length. Responses should occur more than occasionally. Since unautomated spelling can feature rather high item presentation densities per unit time, we ask the system, among other things, to improve on such densities. We do not ask for switching in any exotic sense of speed. As time is measured in 1972, we make the apparently reasonable demand that the down portion of a short down-up "trial" not exceed a few seconds. Below we propose a down-up ratio that is favorable to up.

The video system probably permits the sequence of operations culminating in presentation of a V frame on a CRT to occur in 1 second or less following detection of the controlling digital code under all but the most-impossible heavy-use conditions. The video system can do this even though required to true random access from 1800-frame video storage.

When the day arrives wherein we find it impossible to do contemplated interactive work unless able to true random access from stores the size provided for IDCMS video but characterizing both video and audio, then we probably will need augment the system--e.g., through addition to it of analog-form audio storage that parallels that for video, but without losing the synchronization feature inherent in intrinsic digital codes associated with audio messages. However, the illustrative program requires audio storage at the level of the segment, rather than

at the level of the audio element. Even though the illustrative program is not trivially short, we will argue later that the maximum number of such segments required will be far short of 1800, that audio storage can handle the illustrative program's audio component, and that we can tolerate the relatively slow segment-accessing times so long as the intrasegment interitem down times are of short duration. Given that a segment of interest is in S's audio buffer, we ask the system to step to a following item 1 second following completion of the response to the item that comes before it and to initiate video frame accessing by the end of that second. Assuming, then, a rather large intersegment accessing time, it still would be true, if intrasegment and interitem down times were averaged, that trial down time would average to less than 2 seconds for audio and less than 3 seconds overall. Hence, the illustrative down interval for instructional items is taken as 3 seconds. Acceptability of an interval of this magnitude depends on magnitude of the up interval.

Let audio come up on earphones and video come up on the CRT at the beginning of a 4th second. Audio will go off at the end of the 4th second. Conceivably, there will be younger children who could perform a 3 or 4 letter imitation response through the keyboard during 5th and 6th second. If so, then these fastest children reflect a worst case (a 3-sigma above the mean case). A median-paced child more likely will use 4 seconds (5th through 8th) to complete such responses (under system encouragement at that). Hence, we might think of the presentation-response interval as being 3 seconds under worst-case instructional conditions and 5 seconds on the average. Thus, worst-case item consumption rate for AV items will be 10 per minute (per terminal). The average item consumption rate for such items will be 7-8 per minute (per terminal).

Testing features AO mode presentation. Hence, V retrieval time drops out during testing, reducing down time to 2 seconds. If the argument is correct that the spelling response should require more time than the imitation response (I am not sure that it is), then the second saved from down time might be used during testing to prolong the up interval. (There is little point in relaxing A item retrieval time during testing unless it can be shown that different subsystems might profitably be employed, depending on whether AV or AO mode applies.)

Instruction probably will on occasion be punctuated with little speeches. Such speeches should improve the down-up ratio; whether they would do so in an instructionally productive sense is unknown.

The picture sketched above yields a worst case 1:1 ratio for AV and 1:2 ratio for AO; it yields an average-case 1:1.67 ratio for AV and 1:3 ratio for AO. (Should one wish to evaluate probable 6-terminal traffic under normal distribution, a 3-sigma below the mean case could be viewed as using three times as much presentation-response time as worst case.)

Were we five years into the future--that is, if pedagogical implications of pacing factors had been pretty well established in consequence of putting IDCMS to extensive use--then it might be agreed that ratios much less favorable to up time might suffice. However, IDCMS is the instrument one will use to evaluate such matters. We cannot test toward the upper limits for speeded instruction if hardware is allowed to limit the empirical attack so as to preclude such tests.

Foregoing remarks establish desirability of a worst-case item consumption rate of 10 per minute, referenced to one terminal. Magnitude of the storage requirement underlying support of such a rate remains to be determined.

Random Access Requirements. When one has a large collection of items any one of which one may wish to display at any time, then true random accessing to such items is required. If items of the collection do not class under a mnemonically-useful linear or hierarchical taxonomic scheme, then the sort of storage we see in Video Master Disc becomes required. Even if mnemonically-useful classification is possible, it has no useful implications for Video Master Disc storage when this is used in connection with IDCMS terminals, because multiple usership of IDCMS precludes a movable head staying in one place over item presentations referencing to a given S. Thus, it would be no advantage to group video frames physically on a disc just because they are likely to be used in the same segment of instruction or even in a fixed order in that segment.

When we transfer over to Audio Master Reproduction (the audio store), it again becomes difficult to see how a classification scheme could aid equipment response time per user under the condition of true random accessing from a large number of alternatives. However, the audio tape and video subsystems differ in one important respect. Audio Buffer can receive a sequence of audio elements that is physically grouped in Audio Master Reproduction (although I am not sure that such a fixed-sequence segment then can be made to run as the audio portion of a series of AV presentations whose pace is controlled by the system, at least not until different software is provided than apparently currently is contemplated by the contractor). Video Buffer can only receive one frame at a time. Two sorts of classification for the audio system might prove useful.

Assume first that children will be at or near the same point at the outset of the instructional program but at different points later on. To discourage different users entering (or wishing to enter) the same tape transport of Audio Master Reproduction at the same time, this might necessitate having 2-3 copies of the "first quarter" of the program available in Audio Master Reproduction at the outset--that is, 4-6 transports might contain one copy of the first quarter of the program (where 12 transports are available). In consequence, during this phase of instruction 2-3 terminals might be tied to the 4-6 transports containing one copy of the instructional segments for the phase. Queuing to reach

the same or adjacent segments would only be a problem at the outset unless Ss were homogeneous for pace. Following an intial phase, the whole-copies feature of storage would drop out. Thereafter, the different transports might be used in linear organization to reflect the most-useful linear organization of remaining segments. Whether early or late, segments might be stored according to a system which minimizes predicted queuing to audio tape transports.

Assume second that most segments of instruction can be assembled off-line ahead of instruction in consequence of fully-explicated design-development specifications. That is, items of a segment can be ordered without respect to performance data and, if alternative orders are required, then differently-ordered but otherwise identical copies of the segment can be assembled, again without regard to performance. This need not mean that a segment must run its course. The segment could be terminated at any point in consequence of S reaching an upper or lower criterion bound defined through application of sequential analysis. The important questions then become a) How many audio sequences and reordered copies of these sequences would inhere in the largest portion of the program that needs to be stored on a worst-case day? b) Will we ask the system to assemble such sequences (when properly instructed) on-line?

Earlier consumption rate assumptions warrant the view that Day 10 will be the worst-case day for illustrative instruction. At the end of Day 10, the fastest child will complete the program, the average child will be 2/3 through, and the slowest child perhaps 1/3 through. Hence, worst-case storage is 2/3 of the program (although without whole-program duplication).

The amount of audio sequence storage determines system response speed when the audio tape system is used. If we assume sequences will run on the order of 2 minutes in the worst case, then we might well be willing to tolerate the almost-20 second rewind time it takes any tape transport to reach out to either end of its tape and to transfer 150 inches of storage to Audio Buffer--a worst case for storage position. What remains then to determine is how many such sequences will occur in the terminal 2/3 of the program.

If we define the central instructional path on the fastest child, then it will be 600 minutes long, including central instructional path treatments and loops. Let us imagine that this child will loop one time in five. The 600-minute program yields 300 2-minute segments. Of these, the 4:1 ratio between mainline and loops apportions 240 segments to mainline and 60 to loops. Let us interpret this as 120 instructional segments, 120 test segments, 15 instruction-referenced diagnostic segments, 15 instruction-referenced prescriptive segments, 15 test-referenced prescriptive segments, and 15 alternative test segments. Let us imagine that the average child will show a 3:1 ratio between mainline and loops, necessitating apportioning 80 segments to loops. Further, let us imagine that the slowest child will show a 2:1 ratio between mainline

and loops, necessitating apportioning 120 segments to loops. Hence, the fastest child will negotiate 300 (240 + 60) segments, the average child 320 (240 + 80), and the slowest child 360 (240 + 120). The slowest child defines worst case for storage.

We indicated above how much looping we will assume children will do, rather than the looping segment storage requirement, which is a different thing. For present purposes, let us assume that the slowest child's diagnostic, prescriptive, and alternative test segments are drawn from a pool of segments that is twice the size used--240 segments. The unused segments (which might be used by other users) are there because we cannot be sure just what segments will be needed. If several sets of users revealed that some are never used, they would be deleted. (Conversely, experience might also show that new segments were needed.)

On the worst-case day (Day 10), the slowest child is approximately 1/3 through the program and the fastest child is completing the program. Assuming that the looping level is constant across the program--probably a worst case--then on Day 10, the system would need to store just in excess of (2 x 480)/3, or 320 segments of digitally-coded audio material. Let us assume 20 digitally-coded audio elements per segment plus whatever is needed to address the audio portion of the segment. Let us assume that each segment can be stored on approximately 170 inches of tape (45 seconds x 3.75 ips, with digital codes occurring ahead of their audio elements).

Based on the classification assumption made earlier, it is assumed that two students never will go to the same Audio Master Reproduction tape transport at the same time. S's own audio buffer contains two tape transports. We assume here that while one of these tape transports is in use, the system always will anticipate the most-probable next segment and cause it to be delivered to the other tape transport before the playing segment is completed. At worst, then, the system will sometimes be wrong. Only in those instances will interactive instruction suffer something on the order of a 20-second worst-case rewind-transfer-record-rewind penalty. Such a penalty should be encountered no more than 1 time in 3 for the slowest child and 1 time in 5 for the fastest child, even if the system were operating on the basis of chance rather than on the basis of prior performance by S, including on the segment being negotiated. To summarize, the mean such penalty would be approximately 10 seconds, experienced one time in four (or less). No problem here.

Audio Master Reproduction consists of 6 modules, each containing 2 tape transports. Each transport is reel-centered on 490 feet of tape whose normal-play speed is 3.75, with bidirectional drive over either of two tracks. Each transport will take 52 minutes of audio material, which it will high-speed search at 150 ips. Each track in either direction from reel-center measures 2940 inches. Taking the segment as 170 inches long, each track in either direction can store approximately

17 programs. A single tape transport, then, can carry on the order of 65-70 segments. Since AMR contains 12 tape transports, half of AMR storage will accommodate the worst-case day storage requirement (320+ segments).[3]

It appears possible that placing a large number of small program segments on a tape might necessitate minor modification of system hardware or software. The system typically employs a "2 program" per direction per track view of audio storage in AMR. Whether this is merely for expository purposes or reflects a system design commitment remains to be determined.

If a segment can be loaded into Audio Buffer and if the Buffer then can be made to start, pass over and relay the digital code in approximately one second, stop while video catches up, then pass over and transmit the audio element in synchrony with video, then stop until an instruction from the system controller turns off video, then start through the next item, then perhaps the sketched instructional program is consonant with the audio tape subsystem. If that is so, then the problems referencing to number of stored audio "items" and accessing time give way to other problems that research software design must address. While we have hinted at the magnitude and forms of such problems in the illustrative sketch, they are not central to the present paper and so will be handled elsewhere.

Some of the segments discussed above are AO, rather than AV. These seem less a problem than AV, so warrant no special comment. (Whether VO segments, not of interest here, constitute a problem depends on whether a masked 55 Hz subcarrier is audible when unmasked. It depends on amplitude of the subcarrier.)

Staff probably would not take kindly to closing out on a high-speed retrieval capability referencing to a large collection of items in audio form. Conceivably, careful study will reveal a need to augment any such program as has been sketched with a capability for retrieving one or two items from a large collection all of whose items are quickly retrievable. While the casual treatment presented above does not clearly reveal such a requirement when the equipment is under interactive research use, further study might show the research need for it; moreover, other-than-research uses clearly imply such a capability. Hence, my inclination would be to pursue the question of an audio subsystem that can handle as many "short" audio elements as the analog-form video

---

[3] It is not clear from available contradictory materials whether transport storage should be 52 minutes or 26 minutes. Assuming the 26-minute figure, then total AMR storage would need be used on the worst-case day.

subsystem handles--although it remains to determine just how many items such a device should permit high-speed true random accessing to.

Interactive Research

One reason why I believe that IDCMS will increase SWRL's interactive research and instruction capability from tenfold to one hundredfold is that the state of the art base is low. A study now in formulation by John Koehler has follow-on interactive research implications, although they study itself will not require interaction in the sense of prescriptions that lead S off his treatment-defined mainline. One purpose of the Koehler study is to determine what sorts of prescriptive departures from mainline instruction in the phonics domain should occur. Such information, to be obtained during the second half of the current school year, should show the way toward follow-on research each of whose treatments features student-system interaction in the sense of performance-contingent departures from the mainline.

The Koehler study will evaluate each of three entry skills underlying a segmentation-blending level of phonics instruction, will assign S's to one of three training treatments consonant with entry proficiecies, and will retest entry skills (using novel rule words) following training. This sequence will be repeated through three cycles--one each for CVC, CCVC, and CVCC type items. Two training strategies will be used: one that makes the letter-sound rule the basis for segmentation and blending, one that employs bipartite patterns.

CVC entry skills pretesting will be as follows:

1. Test ES-1. 6 items.

   1.1 Illustrative item. A+V = /cæt/+cat, imitate A (10 sec);
   A+V = /bæd/+cat bad, imitate A (10 sec);
   A+V = "Now say this word" + bat in the context cat bad. R = /bæt/ (15 sec).

   1.2 Maximum time. Each item presentation will involve presenting three different AV pairs. The first two of these will require imitation responses in oral form, the third, a test response in oral form. E will evaluate these responses. If the response is made in less time than is shown above, E will make a keyboard response that advances the ES-1 segment. If no such response is made in the time allowed, the system will automatically step the ES segment to the next AV pair. Maximum time is: 5 sec (ES-1 segment retrieval + duplication) + 6(3)(2) sec (down time) + 6(35) sec (up time) = 251 sec. = 4.5 min.

2. Test ES-2. 6 items.

   2.1 Illustrative item. VA = m-at bat, "Say the second word;"
   R = /b/+/æt/ (15 sec).

   2.2 Maximum time. Same procedure as 1.2. 5 sec (ES-2 segment retrieval + duplication) + 6(2) + 6(15) = 107 sec = 2 min.

3. Test ES-3. 6 items.

   3.1 Illustrative item. AV = /b/+/æt/, "Say as one word;"
   R = /bæt/ (15 sec).

   3.2 Maximum time. Same procedure and time as 2.2. 2 min.

Both above and below, only segments, segment compositions, E override on interval length, and time durations are meant to be definitive even in the illustrative sense. Finer details remain to be specified (and at any rate are unimportant here). The assumption is that ES-1, ES-2, and ES-3 segments for CCVC and CVCC cycles will have the same system-burdening characteristics as are shown above for CVC entry skills pretesting segments.

Training will emphasize analysis, synthesizing, or both, depending on entry skills pretest outcomes. Koehler and his associates distinguish between 5 training subskills appropriate to proficient word attack of the segmentation-blending variety. If entry skills proficiencies indicate an analytic deficiency then Subskills 1-3 are germane (or so it is assumed); if pretesting indicates synthetic deficiency, then Subskills 3-5 are germane (or so it is assumed); if deficiencies of both sorts are indicated, then all of the subskills may need be addressed during training. Koehler indicates that all children will receive training directed to all subskills, but that the emphasis will vary depending on the hypothesis reached in consequence of pretesting. Thus, for example, an S hypothesized to be deficient in analytic skills would have Subskills 1-3 emphasized and Subskills 4-5 treated more perfunctorily. Because all training treatments feature Subskill 3--which uses the most time--for illustrative purposes we simply will assume that those suspected of being analytically deficient will receive essentially no time on Subskills 4-5 and those suspected of being synthetically deficient will receive essentially no time on Subskills 1-2. This permits spreading the Ss of different treatments in such a way as to more-realistically burden the system.

CVC training for the combined analytically-synthetically deficient is as follows:

1. Train 1. 2 items x 2 presentations = 4 trials.
   AV = /pæt/+pat; imitate A. (Illustration)
   Down = 5 + 4(2) = 13 sec, up = 4(10) = 40 sec, total time = 1 min.

2. Train 2. 2 items x 2 presentations = 4 trials.
   VA = pat + /p/+/æ/+/t/; imitate A. (Illustration)
   Down = 5 + 4(2) = 13 sec, up = 4(10) = 40 sec, total time = 1 min.

3. Train 3. 6 items x 5 presentations = 30 trials.
   VA = a + /æ/; imitate A. (Illustration)
   Down = 5 + 30(2) = 65 sec, up = 30(3) = 90 sec, total time = 3 min.

4. Train 4. 2 items x 2 presentations = 4 trials.
   VA = m-a-p + /m/+/æ/+/p/; imitate A. (Illustration)
   Down = 5 + 4(2) = 13 sec, up = 4(15) = 60 sec, total time = 1.5 min.

5. Train 5. 2 items x 2 presentations = 4 trials.
   VA = m-a-p + "Say the word;" R = /mæp/. (Illustration)
   Down = 5 + 4(2) = 13 sec, up = 4(15) = 60 sec, total time = 1.5 min.

Again, we assume that CCVC and CVCC cycles will have the same system-burdening characteristics for Train-1 through Train-5 as does the CVC cycle. Training materials will not be the same for letter-sound and bipartite patterns; the same items will be used but segmented and blended differently in terms of units used. For the present we assume that the different strategies will be consonant with training coverage illustrated above for the letter-sound pattern strategy.

Posttests will test for the same skills as do entry skills tests except that novel rule words will be used. Because the letter-sound rule set will consist of only 6 items after Cycle 1 training, increasing to 12 items after Cycle 2 and to 18 items after Cycle 3, possibilities for generating novel rule words will be meager following Cycle 1 training and greater with completion of training on succeeding cycles. Hence, it is assumed here that there will need be fewer items in the CVC posttests than in later ones. While this could be remedied by teaching the most rules during Cycle 1 and the least rules during Cycle 3, there is no way to approach the experimental design problem that is entirely satisfactory. Here it is assumed that Cycle 1 posttests each will contain 4 items, Cycle 2 posttests 7 items, and Cycle 3 posttests 10 items. The assumption is that posttests will use the same amount of time per item as do entry tests.

The fixed segment view typically would not sit well with E, particularly for testing segments but, to a lesser extent, also for training segments. Before determining how many alternate random-ordered versions of a segment might be employed, let us inventory segments. There will be 3 ES test segments and 3 posttest segments for each cycle, or 18 different test segments. There will be 5 training segments for each cycle for each strategy, or 30 different training segments. Since segments will be of different lengths, it is necessary to calculate audio storage based on segment lengths. We will assume that maximum times given above entail that amount of audio storage, which usually will not be so. The 3 3-segment ES tests will cost 8.5 minutes each, or 25.5

minutes if stored together (or at the same time). The 3 3-segment posttests will, respectively, require approximately 6, 9, and 12 minutes, or 27 minutes if stored at the same time. The 6 5-segment training segments each will require 8 minutes, or 48 minutes if stored at the same time. Assuming 26 x 12 minutes of audio storage, or 312 minutes in all, then if only a single version of each needed segment were used with all versions stored in the audio store throughout the study, just under 1/3 of total storage would be used. Under the circumstances, E could afford 3 versions of each segment (or, if the tape transport capacity figure of 52 minutes is used, 6 versions of each segment). The research format probably precludes the degree of spreading that the instructional illustration presented earlier invites. If the worst-case day necessitates storing 1/3 of the segments needed during the life of the study, then 26-minutes storage per tape transport is consonant with 9 randomly-ordered versions of each segment (and 52-minutes storage with 18 versions).

Foregoing remarks suggest that maximum times for analytically trained groups (2 groups, each N = 12), by cycle, will be 19, 22, and 25 minutes, or 66 minutes in all. Maximum times for synthetically trained groups (2 groups, each N = 12), by cycle, will be 20, 23, and 26 minutes, or 69 minutes in all. Maximum times for combined analytically-synthetically trained groups (2 groups, each N = 12), by cycle, will be 22, 25, and 28 minutes, or 75 minutes in all. Assuming use of 20-minute periods only 15 minutes of which was usable, then under the conditions of 6-terminal usage of IDCMS, one E per terminal, and good administrative support regarding scheduling of Ss, the study as illustrated could be accomplished using the system 4 hours per day (9-12, 1-2) for 5 successive school days. Assuming appropriateness of the character generator for all video elements, the study materials could be constructed without recourse to artists or other printshop personnel. If we made it a requirement that the system (and augmenting equipment) be used for purposes of constructing materials only during normal down time (e.g., before 9 a.m., and after 4 p.m., during lunch hour, on weekends), then perhaps a single technician could construct, organize, and load in everything needed to support a study of the sort outlined above in 8-16 hours without hindering system exploitation for research purposes. Doubling that figure to provide for checking and editing and doubling it again to accommodate study-tailored production of software (if that proves needed), the system might need to be used on the order of 48 hours during its down time for purposes of supporting study execution and 20 hours for purposes of executing the study and performing whatever analyses were required. The personnel bill--exclusive of study formulation and write-up--would be 48 technician hours, 120 research assistant hours, and perhaps 20 IDCMS-supervisor hours. My prediction is that this would represent at least a fivefold savings on what the study actually will cost when conducted using mobile laboratory facilities at the schools and pre-IDCMS capability for providing graphic and analysis support. We need to do better than that; I suspect that we would do appreciably better the second time around.

It is much easier to make the case for a high-storage, short-element, high-speed audio accessing subsystem when the task is to obtain experimental findings than when it is simply to tryout-evaluate instruction. While the latter task for the most part can be accomplished with fixed-order segments, experimental tasks almost always require that alternate orders be used to insure that an order bias is not operating. There do not appear to be so many different short audio elements in the Koehler study as outlined above--a rough calculation suggests that there are between 200 and 250 such elements. However, it would be quite easy to increase this number 4 to 8 times in a slightly more ambitious study of essentially the same sort--e.g., by extending cycles to all possible one-syllable word forms, all pertinent letter-sound rules, and all novel rule words consonant with these extensions. Given high-speed audio accessing, segment formation and randomization could be placed on the synthesizing basis that precludes appreciable investments in forming segments and alternate versions of segments (or storing them in segment form). In that event, software would control segment formation and randomization. Whether this would mean simply that the materials construction savings would be obtained for a like (or greater) dissavings in programmer time should be evaluated.